

REMARKS

Claims 1, 4-7, 24-25, and 32-33 have been amended. Claim 3 has been cancelled. No new claims have been added. Thus, claims 1, 4-7, 24-25, and 32-33 are pending.

The specification stands objected to under 35 U.S.C. § 112, first paragraph. Claims 1, 3-7, 24, and 32 stand rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the written description requirement. Claims 1, 24, and 32 have been amended and it is now believed that the specification and the claims are in full compliance with 35 U.S.C. § 112, first paragraph. Accordingly, the objection to the specification and the rejection to the claims under 35 U.S.C. § 112, first paragraph, should be withdrawn.

Claims 1, 3, 24, and 32 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Record (U.S. Patent No. 5,335,484) in view of Lehr (U.S. Patent No. 5,989,873) and Song (U.S. Patent No. 6,061,711). Claims 4 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Record, Lehr, and Song in view of Borkenhagen (U.S. Patent No. 6,212,544). Claims 5-6 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Record, Lehr, and Song in view of Ellis (U.S. Patent No. 5,088,036). Claim 7 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Record, Lehr, and Song in view of Toutonghi. Claims 25 and 33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Record in view of Toutonghi (U.S. Patent No. 5,842,016). These rejection are respectfully traversed.

Claim 1 recites, *inter alia*, "A program control apparatus ... comprising: a first means for setting the prescribed flag to a first state, ... responsive to a first application program interface call for requesting a start of context switching detection made from a first thread of said plurality of threads; a second means for setting the

prescribed flag to a second state, ... responsive to context switches and sets the prescribed flag to the second state when a context switch is detected while the prescribed flag is in the first state; and a means for producing a first return value when the prescribed flag is in the first state and for producing a second return value when the prescribed is in the second state, wherein said means for producing a return value is responsive and passes said return value to, said first thread making a second application program interface call requesting a termination of context switch detection; and a means for invalidating said first thread if the return value is set to said second value when said first thread calls said second application program interface."

Claim 24 recites, *inter alia*, A method ... comprising the steps of: in response to a first application program interface call from a first thread from a plurality of threads which requests a start of context switching detection, setting a flag to a state indicating absence of context switching; in response to a detection of context switching after said flag is set to the state indicating absence of context switching, setting said flag to a state indicating presence of context switching; and in response to a second application program interface call from said first thread, providing a return value indicating the state of said flag to said first thread; and invalidating said first thread if the return value corresponds to said flag indicating presence of context switching."

Claim 25 recites, *inter alia*, "A method ... comprising the steps of: in response to a first application program interface call from a first thread from a plurality of threads which requests a start of detection of a data write to a designated memory area, setting a flag to a state indicating the absence of a data write; in response to a detection of a data write to the designated memory area after said flag is set to the state indicating the absence of a data write, setting said flag to a state indicating a presence of a data write; and in response to an second application program interface call from said first thread, returning a value corresponding to the state of said flag to said thread; and

invalidating said first thread if the return value corresponds to said flag indicating presence of a data write."

Claim 32 recites, *inter alia*, "A computer readable recording medium storing a program ... said program ... includes the steps of: in response to a first application program interface call from a thread which requests a start of detection of context switching, setting a flag indicating the absence of context switching; in response to a detection of a context switch, and after said first application program interface call, setting said flag a state corresponding to presence of context switching; in response to a second application program interface call from said thread which requests a termination of detection of context switching, returning a value corresponding to the state of said flag to said thread; and invalidating said thread if the return value corresponds to said flag indicating presence of context switching."

Claim 33 recites, *inter alia*, "A computer readable recording medium storing a program allowing a computer to execute a program control method, said program control method comprising the steps of: in response to an first application program interface call from a thread which requests a start of detection of a data write to a designated memory area, setting a flag indicating an absence of data writes; in response to a detection of a data write to said designated memory area, setting said flag to a state corresponding to presence of a data write; in response to an second application program interface call from said thread which requests a termination of detection of a data write to the designated memory area, returning a value corresponding to the state of said flag to said thread; and invalidating said first thread if the return value corresponds to said flag indicating detection of a data write."

Record discloses an event monitoring system for a computer system which permits a thread to define, via a first API, a type of event to monitor. When an event of

the type being monitored occurs in the computer system, a signal is sent to an event handler, which stores the event. The stored event can be subsequently reviewed using an event monitor. Column 2, lines 40-57.

The Office Action alleges that limitations directed to invalidating a thread are taught by Record. However, the portion cited by the Office Action merely states:

Still another attribute in the event monitor definition specifies whether or not other program threads within the same process as the event handler should be suspended after the event monitor is satisfied and while the event handler receives the event notification and handles the event. It may be important to suspend execution of the other program threads if the event is a system failure which requires an abend or an event which requires the event handler to take a "snapshot" of the entire process.

Record, Column 9, lines 37-46.

The above recited passage merely states that other threads can be suspended (i.e., temporarily halted in such a way that they can be subsequently resumed at the point where halted) so that a snap shot can be taken. Each of the above recited claims requires invalidation of a thread (i.e., termination of the thread such that it cannot be resumed at the point where halted). Indeed, there is no disclosure or suggestion in Record regarding performing a thread invalidation, as Record is directed to an event monitoring and monitoring events does not require, and is inconsistent with, invalidating the thread which may execute the event being monitored. In this manner, Record actually teaches away from the above recited portions of the independent claims.

The Office Action additionally cites to Lehr, Song, Borkenhagen, Toutonghi, and Ellis. However, these references, whether taken singly or in combination, discloses the above recited features of the independent claims.

Thus, claims 1, 24-25, and 32-33 are believed to be allowable over the prior art of record. Depending claims 4-7 are also believed to be allowable for at least the same reason as claim 1.

In view of the above, each of the presently pending claims in this application is believed to be in immediate condition for allowance. Accordingly, the Examiner is respectfully requested to pass this application to issue.

Dated: March 3, 2004

Respectfully submitted,

By 

Thomas J. D'Amico

Registration No.: 28,371

Christopher S. Chow

Registration No.: 46,493

DICKSTEIN SHAPIRO MORIN &
OSHINSKY LLP

2101 L Street NW

Washington, DC 20037-1526

(202) 785-9700

Attorneys for Applicant